Latency improvement in sensor wireless transmission using IEEE 802.15.4

Emmanuel Fléty, Côme Maestracci

Latency improvement in sensor wireless transmission using IEEE 802.15.4

Emmanuel Fléty, Côme Maestracci
IRCAM - Real Time Musical Interactions
STMS IRCAM - CNRS - UPMC
1 Place Igor Stravinsky
75004 - Paris - France
emmanuel.flety@ircam.fr

ABSTRACT
We present a strategy for the improvement of wireless sensor data transmission latency, implemented in two current projects involving gesture/control sound interaction. Our platform was designed to be capable of accepting accessories using a digital bus. The receiver features a IEEE 802.15.4 microcontroller associated to a TCP/IP stack integrated circuit that transmits the received wireless data to a host computer using the Open Sound Control protocol. This paper details how we improved the latency and sample rate of the said technology while keeping the device small and scalable.

Keywords
Embedded sensors, gesture recognition, wireless, sound and music computing, interaction, 802.15.4, Zigbee.

1. INTRODUCTION
The use of wireless sensor data transmission has been used for several decades in computer music either as a sensing technique by itself [8][11] or as digital transmission media [3][1].

Wireless solutions are now even widely used in interactive gaming with devices like the Nintendo Wii remote or Sony Playstation Move controller.

Those ready-made devices can be extensively hacked and modified to serve the purpose of musical interaction, however they suffer from a clear lack of performance in the sensor sampling rate and/or the latency jitter, two aspects leading to a major impact on the interaction quality [10].

Our goal was to improve our current wireless system that uses off-the-shelf Xbee zigbee modules (presented in previous NIME[2]). XBee modules are popular and feature suitable performance, allowing multi-node sensor digitizers as found in the Sense/Stage Project[7]. However, their main drawback is a locked firmware which does not allow a seamless configuration of each node. We therefore specifically redesigned our hardware into a more generic platform for enhanced performances. This research was performed in the context of *Interlude* and *Urban Musical Game* projects where gestural interface were built for either collaborative music playing or music pedagogy [9].

NIME'11, 30 May–1 June 2011, Oslo, Norway.

2. KNOWN ISSUES
Designing a wireless system presents several known issues and challenges such as CSMA/CA when the media is shared amongst several nodes (star or mesh networks) and the implementation of the numerous layers of the OSI model.

Turnkey radio modules are affordable and easy to implement with custom electronics including usually a small microcontroller unit (MCU) acquiring the sensors' data. However forwarding the data to the radio module has an important impact (slow UART) on the system latency which could be significantly improved.

Widely used for experiments, single radio channel transmitter/receiver pairs (wireless UARTs) were legion a decade ago with the known drawback of permanently occupying the radio channel, making the simultaneous use of multiple nodes on stage impossible without using several radio channels which were highly dependant on local FCC regulations.

In order to allow multiple performers on stage, we therefore oriented our past designs toward actual networking technology such as 802.11 [4]. This was made easier thanks to the emergence of the worldwide 2.4 GHz ISM band for general purpose broadcasting and dedicated wireless networking which was unfortunately bulky and featured a limited runtime.

Bluetooth appeared as a promising, power friendly solution. It is still widely used for hacking but it has not received a positive echo for live performance use despite a fairly high data rate as pointed out by Torrensen [12]. Bluetooth suffers from a significant jitter in the transmission latency[1] and its packetizing effect results in the loss of the original timing of the sampled data. This usually worsens when a Bluetooth module is directly paired with a host computer since the software Bluetooth stack is not designed for real-time applications and features unmanageable time-outs, making the wireless link impossible to reset safely during the performance.

To overcome these issues, we oriented our subsequent designs towards the IEEE 802.15.4 standard, a simplified, reduced power consumption, sensor network oriented, wireless protocol. Its implementation will be discussed in the next sections.

Section 3 will highlight the limitations of our previous systems and the general bottlenecks we are trying to overcome. Sections 4 and 5 will detail our proposed improvements. The 6th section will discuss experimental results.

[1] Author measured up to 30ms on SSP Bluetooth units.

3. GENERAL ARCHITECTURE

3.1 PHY and MAC implementation

The IEEE 802.15.4 is a standard which specifies the physical layer and media access control for low-rate wireless personal area networks (LR-WPANs) [12].

It operates mostly on the 2.4 GHz (ISM) band and features a on-air raw data rate of 250 kbps. Once the MAC layer has been implemented, the actual useable bandwidth for user data drops to 101 kbps [6].

One important bottleneck limiting the data rate is the communication between the sensor acquisition system and a wireless unit such as a XBee OEM module [13]. It is frequently achieved most with a UART which tends to be slower than the wireless data rate therefore adding to the overall transmission latency.

As an example, 6 sensors are sampled on 10 bits (2 bytes) and transmitted over a 115200 baud UART take 1.041 ms to reach a Xbee module.

This duration has to be compared with the maximum duration of the radio transmission [6] :

$$t_{transmit} = t_{worst\ case\ channel\ access} + t_{frame\ transmission} + t_{turn\ around} + t_{ack}$$

For the 12 byte payload above :

$$t_{transmit}\ (ms) = 2.368 + 0.576 + 0.192 + 0.352 = 3.488\ ms$$

Therefore, the additional transmission from the acquisition unit to the wireless module adds dramatically 30% of latency to the whole system.

3.2 Sensor sampling

Sampling analog sensors using the shared ADC of the microcontroller unit increases the used CPU, even if handled with interrupts. Most embedded sensor hardware cannot afford the proper analog front-end that would improve the multiplexer switching time because of the room it requires.

The obtained slew-rate relies the internal clock scheme and the sensor current sourcing. An average value of 120 µs acquisition time can be easily obtained with a maximum of 1 LSB of cross-talk between channels (16 MHz PIC microcontroller). Our 6 DoF sensor would require at least 720 µs of acquisition time using that sampling method.

3.3 User data protocol (OSI layers 5-6-7)

Formatting and translating sensor data is also a source of latency. To avoid adding latency, the system must have a coherent data encapsulation scheme and avoid packet translation to minimize impact on the system performance.

4. IMPROVED DESIGN

4.1 Hardware

Our previous design used a Xbee Zigbee module which stacks its protocol over 802.15.4. with no user access.

In order to virtually suppress the MCU to MAC/PHY latency discussed above and to author our own firmware and packet protocol, we opted for a microcontroller featuring 802.15.4 internally. A combination of hardware and software makes the data transmission between the user program and the MAC layer as fast as a RAM transfer. Data is further shifted out through the QPSK radio modem by the internal hardware.

We use a Jennic 5139R1 OEM module [14] that embeds a 16 MHz, 32 bit RISC microcontroller, 96 KB of RAM, a 802.15.4 MAC software and hardware stack associated to a 2.4 GHz radio as well as several handy peripherals to interface with sensors and external devices (I^2C and SPI digital interface, UART, GPIO, ADC, DAC and comparators).

The Jennic MCU is programmed in C language. Low level hardware access is eased with API functions while the 802.15.4 stack is proposed as software template. The module is powered by a 3.7V Li-Po battery cell and embeds its own charger.

4.2 Sensors

To reduce the ADC sampling scheme we use pre-digitized sensor read using the I^2C bus.

Our sensor node is composed of an Analog Device ADXL345 3D accelerometer and an Invensense ITG-3200 3D gyroscope. Each node can be extended with more sensors using either I^2C compliant digital sensors or optional accessory boards translating classic analog sensors to I^2C.

4.3 Communication with the host computer

A base station was developed as a WPAN network coordinator using also a Jennic MCU. The base station cannot really be considered as just a receiver since it achieves communication both ways, just like the sensor nodes.

Popular solutions such as Arduino use a serial port to send data to the computer, sometimes over a SLIP socket. While easy to implement on small MCU, serial links add to the latency of the system ; in order to keep the data path as fluid as possible and avoid further bottlenecks, we used ethernet to communicate with the computer, this time using a Wiznet812 100BASE-T module rather than our former 10BASE-T solution, therefore dividing the corresponding latency by a factor 10. Moreover, ethernet allows up to 100 m long data links to the computer.

Data is exchanged with the host computer using the OpensSoundControl (OSC) protocol over UDP, allowing easy data parsing as well as up-link configuration parameters that can be sent to the sensor nodes as discussed in section 3.4.

The Datagram contents is sent from the Jennic MCU to the Wiznet module via a 8 MHz SPI bus ensuring a inter-component high speed data exchange.

Finally, the base station is configured using a web server hosts and a parameter page accessible in a web browser.

4.4 Protocol and Services

Our goal was a generic sensor node capable of accepting sensor *accessories* as illustrated in figure 4. Specifically, this corresponds to extend the node with external daughter boards. There is no hardware dependency coded in the firmware aside the I^2C driver of the two 3D onboard sensors.

On top of this direct sensor access, we developed a service oriented protocol that allows extra peripherals to be discovered and read by the sensor node. The sensor node is designed as a data collection hub : the hardware dependencies are located in the sensor itself that then communicates with the hub over a high speed I^2C bus using an intermediate 16 bit PIC24F64GA004 microcontroller per accessory. This might also benefits from some local sensor processing (filtering, triggering, sample rate control).

This topology allows for the implementation of various sensor interfacing, such as analog sensors, piezo microphones, matrixed keypads or digitized SPI sensors. I^2C sensors can be used too since the PIC MCU has two I^2C ports. The low level I^2C driver is coded in the intermediate MCU and forwarded over our frame oriented I^2C exchange protocol within the sensor node.

Each accessory, as well as the onboard sensors can have several services which are proposed by the sensor node during the discovery phase. Most of the accessories we designed have

a streaming service with controllable sample rate but can also have high-level pre-processing such as Kalman filtering, onset detector or others algorithms for which an accurate sampling rate is mandatory for proper results. Interfaced as an accessory, continuous analog sensors can be turned locally turned into triggers without the need of a continuous streaming. As a result, more sensor nodes can be used simultaneously in the network without degrading the bandwidth.

Each service sends data frame containing the frame type tag (ints or floats). Each sensor node is identified by a hardware ID installed into the module FLASH using its serial port. The hardware ID is used in the OSC address scheme to route each module data on the host computer :

```
/<hardware ID>/<service #>/ data list
```

All radio communication, MAC addresses and higher level addressing scheme is transparent for the user. The base station WPAN coordinator handles node association and de-association on its own using our implementation of a simplified ARP table. The latter is periodically broadcasted to all nodes on the network to keep each node aware of the network population.

Finally, each radio frame receive a 1 byte packet number to detect short term data drops and the base station uses a sub-millisecond timer to date each packet exported in OSC.

5. IMPLEMENTATION

Since the sensor node is a scalable platform, we were able to use the same hardware base for both current projects.

For the Interlude project, the sensor node took the shape of the "MO" handheld unit accepting plug-in accessories as described in 4.4. The figure 1 show examples of accessories containing daughter boards as described in 4.4.

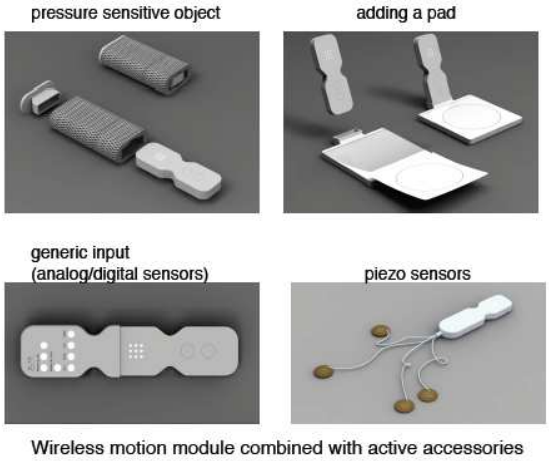

Figure 1. MO configuration array for the Interlude Project (design by NoDesign)

In its smallest form factor (mini-MO), the unit features only the onboard 3D sensors, no led array nor accessories and it becomes a 50x30x13 mm wearable unit rechargeable over a USB port.

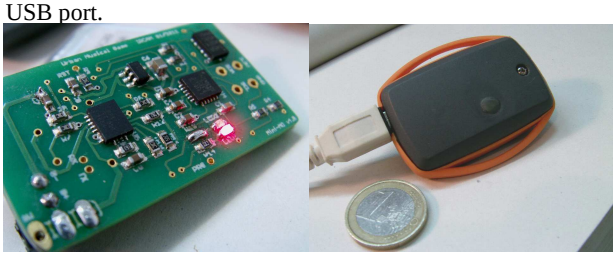

Figure 2. The Mini-MO configuration

6. SYSTEM EVALUATION

6.1 Internal timings

We measured the key acquisition timings using an oscilloscope and a GPIO of the Jennic MCU, setting the GPIO at the beginning of the function call and clearing it at the end of the process.

6.1.1 Data transfer to the MAC/PHY layers

Our final packet formatting is achieved by adding 8 bytes to the "useful" data payload : the frame delimiter, data type tags, message type, packet number and CRC.

For the same sensors' data payload used in the example in section 3.1 (12 bytes), the 802.15.4 frame transmitting the onboard sensors data takes 64 µs to be sent to the Jennic internal MAC stack. This is the most important improvement compared to a separated radio module communicating using a serial port (16 times faster).

6.1.2 Sensor data retrieval

With a speed of 400 kbps, the I^2C bus allows us to retrieve the 3D acceleration (10 bit sampled) and 3D angular speed (16 bit sampled) in 312 µs, API function calls included. This improves the sensor acquisition time by a 2.3 factor.

6.1.3 Base station latency

The build of the OSC packet and its data transfer by SPI also adds some latency. We measured the lag at two locations of the program using GPIOs. The base station processing adds 1.64 ms. This highlights again even a short, optimized OSC message takes some time to be assembled by embedded electronics [5].

6.2 Overall latency

We also measured the actual latency (best case) from the sensors themselves to the reception of the OSC packet in a computer using Max/MSP and an audio card.

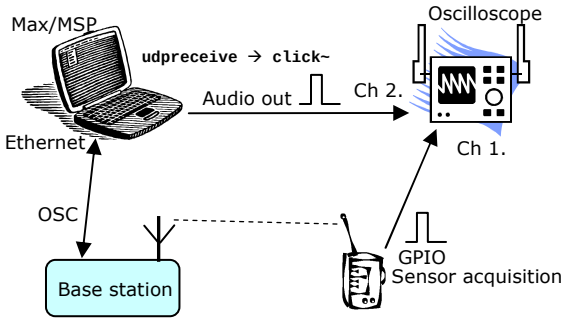

Figure 3. Latency measurement technique

We measure the lag between sensors acquisition and audio click and we kept the shortest duration as a reference.

Total actual latency = measured latency - audio latency

Total actual latency (best case) = 31.8 - 27.6 = 4.2 ms

Using the measurement of 4.1.3 we can conclude the OSC packet latency throughout the operating system and Max/MSP is 400 µs.

Table 1. Latency costs summary for different systems (ms + best case)

	Sensors	Radio	Base station	Host	Total
Xbee Serial	1.25	3.62	7.14	n.a.	12.01
Xbee API	0.8 (est.)	3.62	1.01	n.a.	5.43
Jennic	0.312	1.848	1.64	0.4	4.2

The best case measurements were obtained with no back-off wait from the CSMA/CA channel access. Hence the overall latency expected with our system is {4.2 ; 6.568 } ms.

Using the time stamp generated by the base station, data flow can be re-aligned and lost packets can be detected.

The variable part of the latency is essentially the radio transmission. We experimented a minimum transmission period of the on-board sensors streaming service of 3.2 ms. Up to 3 sensor nodes can be used simultaneously in continuous streaming while staying under the accepted 10 ms range. More nodes can be used with non continuous or asynchronous services.

6.3 Runtime

The Mini-MO version of the sensor node uses an average 52 mA. We use a 290 mAh PCB protected Lithium-Polymer battery pack conferring the device more than 5 hours of continuous use.

7. CONCLUSION

The paper presented how we improved certain aspects of wireless sensors data acquisition using the IEEE 802.15.4 standard. We showed that by using an integrated solution for the sensor node MCU and radio modem, as well as using digital sensors, the system can be 3 times faster than existing solutions using the same radio standard.

While we chose to have an extended radio packet protocol, raw data could be sent hence reducing radio and OSC packets building times resulting of further latency reduction.

The use of the I^2C bus allows a good scalability rather than relying on the number of ADC channels available on the sensor node MCU.

Upcoming work is the design of several sensor accessories for the MO handheld version to build an actual network population in order to evaluate the system performances with a larger number of participants (4 to 8).

The smaller implementation of the sensor node ("Mini-MO") will be used from now as our standard gesture capture unit for dance and augmented instrument projects.

8. ACKNOWLEDGEMENTS

This research is funded by the National Research Agency (ANR) and CapDigital (Interlude project ANR-08-CORD-010), and la Région Ile-de-France.

We would like to thanks the projects consortiums and particularly NoDesign and Da Fact for the design of the MO platform.

9. REFERENCES

[1] Aylward R., Paradiso J., Sensemble: A Wireless, Compact, Multi-User Sensor System for Interactive Dance. In *NIME* 2006.

[2] Bevilacqua, F., Guédy F., Schnell N., Fléty E., and Leroy N., Wireless sensor interface and gesture-follower for music pedagogy. In *NIME* 2007.

[3] Coniglio, M., Stoppiello D., The MIDI dancer. Troika Ranch Web Article - *http://www.troikaranch.org/pubs/ Movement_Research_Paper.pdf*

[4] Flety, E., The WiSe Box: a Multi-performer Wireless Sensor Interface using WiFi and OSC. In *NIME* 2005.

[5] Fraietta, A, Open Sound Control : Constraint and Limitations. In *NIME* 2008.

[6] Jennic. JN-AN-1035, Calculating 802.15.4 data rates. Application note. In *http://www.jennic.com/ files/support_files/JN-AN-1035%20Calculating%20802-15-4%20Data%20Rates-1v0.pdf*

[7] Malloch, J, Wanderley, M. , Sense/Stage - Low cost Open Source sensor infrastructure for live performance and interactive, real-time environments. In *ICMC 2010*.

[8] Mathews, M., The 1997 Mathews Radio-Baton & Improvisation Modes. In *ICMC* 1997.

[9] Rasamimanana, N. & Al., Frechin, J-L., Petrevski, U. Modular Musical Objects Towards Embodied Control Of Digital Music. In *TEI* 2011.

[10] Schmeder, A., Freed, A., Implementation and applications of Open Sound Control Timestamps. In *ICMC* 2008.

[11] Theremin, L., Method of and apparatus for the generation of sound. *US patent 1,661,058* - 5th of December 1925.

[12] Torresen, J., Renton E., Jensenius, A., Wireless Sensor Data Collection based on ZigBee Communication. In *NIME* 2010.

[13] http://www.digi.com

[14] http://www.jennic.com

[15] http://standards.ieee.org/getieee802/download/802.15.4-2006.pdf